

FOUNDATION: TRANSFORMING DATA BASES INTO KNOWLEDGE BASES

R. B. Purves

James R. Carnes
Dannie E. CuttsBoeing Aerospace Company
PO Box 1470, MS JA-68
Huntsville, AL 35807Boeing Huntsville AI Center
PO Box 1470, MS JA-65
Huntsville, AL 35807ABSTRACT

One of the recent problems faced by developers of knowledge based systems is how to best use data already stored in traditional data bases. This paper describes one approach to transforming information stored in relational data bases into knowledge based representations and back again. This system, called Foundation, allows knowledge bases to take advantage of vast amounts of pre-existing data. A benefit of this approach is inspection, and even population, of data bases through an intelligent knowledge-based front-end.

1. INTRODUCTION

A workstation tool (Foundation) has been developed to assist in the complex task of analyzing information contained in engineering data bases. Foundation users are able to graphically display networks of components, query the network for information about the components, and edit data on components in the network. Analysis of the system from various perspectives is supported.

Foundation synthesizes initial structural relationships from the data base data dictionary and is then able to read records from data base tables. Some tables contain structural information, others contain functional information, while still others contain attributive information for particular applications. Structural information is used to construct the initial hierarchical model, then functional and attributive information is used to construct a set of network links between the former data base key values, which are now objects in the knowledge base.

The initial application for Foundation is a knowledge-based interface to Boeing's Space Station Engineering Master Data Base, a system designed for information management in design and development of the NASA Space Station. There are two major key-object mappings: one, part-classes from diverse discipline models describing the many individual components used in the design, and two, part-instances used in describing how all the different parts fit together into an engineering working model. This paper

presents current status and goals of the project, focusing on lessons learned and further application potential.

2. THE WORKSTATION

This development joins two distinct environments, relational data base and knowledge base. The initial work has been in data base environments with well-defined structure and behavioral elements such as those found in engineering design. These elements encompass both static engineering discipline models, (i.e., invariant component structure), and dynamic system models, (i.e., component instance behavior). Translation from the data base to knowledge base system is accomplished through conventional data base access methods driven by the complex qualitative model that is the "foundation" of the knowledge base[3]. The qualitative model used within the knowledge base is the major topic of this section.

2.1. The Data Base

To fully understand the knowledge base model it is necessary to inspect the structure of the typical engineering data base. At high levels the data base is partitioned into tables representing structural or descriptive information, and functional or behavior information. Structural information represents the generic structures of the data base objects, i.e. parts in the data base, as well as the connections between generic parts, such as subcomponent and interface relations. Descriptive information includes mass, thermal, and electrical properties of data base objects. Functional information about data base objects describes the behavior of system components, such as how thermal properties are used to represent conditional behavior or how separate instances of a component function differ within a system.

2.2. The Knowledge Base

In order for Foundation to be used with various data base systems, all access to the data base information is through a query language (SQL) rather than by accessing tables images directly. This approach also helps to ensure data base integrity through the use of the native data base system access and lock protocols.

Structural information is extracted from the data base to define templates of attributes common to all components in a system, as well as component types from which specific component instances are created. Other tables in the data base contain behavioral data about component types. These data are read from the relational data base into the knowledge base in rule form.

The process of translating from the data base to the knowledge base consists of several steps:

1. Read data dictionaries for various tables describing the data base architecture as well as the format and type of data (structural, descriptive, or behavioral) contained within.
2. Establish bounds, (i.e., levels of detail, proper views or perspectives, etc.) on information to ensure that sufficient data are retrieved from the data base. Performance and storage issues drive the need for bounding the data.
3. Read static structures from appropriate data base tables. These structures are used as keys to read static attributive data required for current view or perspective.
4. Read static behavioral data. These data are used to describe the generic behavior of components, but do not vary as the component is used in specific system settings (such as the operating temperature range for a component).
5. Read attributes to describe the instance behavior of a component. The working behavior of a component includes functional descriptions specific to the component and its particular application in the system. Some of these attributes within the description serve only to contain values related to some "working" status, such as current temperature of a fluid in a pump. Some attributes have default initial values set upon creation while others have values assigned during various behavioral inference schemes[1,4].

The architecture of the knowledge model is similar to the data base, with a strict division of generic component and instance information. Structural and descriptive data about components are combined into frame-like structures. However, real differences exist in the complex connections drawn between objects in the instance environment, and in the ability to reason about instances using generic and specific component rules. The component system may be viewed as a hierarchical tree of objects, where each object can be decomposed into its component parts, in a well-connected network of component instances. Figure 1 gives examples of description data contained in the knowledge base.

System functionality is implemented in rule bases, figure 2, where inferences within the knowledge base can be made through forward and backward chaining of component properties or constraints[4]. The resulting inference mechanism, based on propagation of constraints, is applicable to a wide class of physical systems exhibiting discrete or continuous behavior, and can be used with a variety of representations (e.g., quantitative or qualitative)[5].

GENERIC DEFINITIONS:

```
(deftype PUMP-A (generic-type 'PUMP))
(defproperty PUMP-A (pressure (LOW (less-than 0))
                              (NOMINAL (range 0 50))
                              (HIGH (greater-than 50)))
 (temperature (LOW (less-than 10))
              (NOMINAL (range 10 80))
              (HIGH (greater-than 80))))
```

INSTANCE DEFINITIONS:

```
(defproperty PUMP-A ((PUMP-SWITCH-STATUS 'ON)
 (PUMP-STATUS 'WORKING)
 (PUMP-PRESSURE-STATUS 'NOMINAL)
 (PUMP-TEMPERATURE-STATUS 'NOMINAL)))
```

FIGURE 1: Examples of attribute definitions.

GENERIC RULES:

```
(defrule PUMP-WORKING
  IF [AND [IS-A-PUMP =some-pump]
          [INPUT-TO-PUMP =something]]
  THEN (tell [OUTPUT-FROM-PUMP =something])
```

INSTANCE RULES:

```
(defcomponent PUMP 362 'PUMP-A)
(defrule PUMP-SHUTDOWN
  (IF [AND [COMPONENT '<PUMP-362>]
          [PUMP-SWITCH-STATUS '<PUMP-362> 'ON]
          [PUMP-PRESSURE-STATUS '<PUMP-362> 'LOW]
          [PUMP-TEMPERATURE-STATUS '<PUMP-362> 'HIGH]]
  THEN (tell [PUMP-STATUS '<PUMP-362> 'ABNORMAL])))
```

FIGURE 2: Examples of rule definitions.

3. CONCLUSIONS AND FUTURE DIRECTIONS

Early work on Foundation shows the feasibility of transforming data from a relational data base into a knowledge based format. Work of this nature is instrumental in bridging the gap between traditional data base systems and knowledge based systems, allowing knowledge based systems to take advantage of existing information stored in data bases. It also paves the way for storage of the large amounts of knowledge structures necessary for more advanced reasoning systems.

Although this project focuses on the "well-defined" environment found in engineering disciplines, future work will represent engineering information at greater and deeper levels of detail. Finer grained properties will be attributed to smaller entities, while larger collections or systems of these entities will be attempted. Reasoning over time will be accomplished with the coupling of time intervals and properties[5].

A new problem challenging the artificial intelligence community is "design knowledge capture," that is, the construction of systems with inherent built-in evolvability toward more advanced technologies and machine intelligence[2]. Knowledge within these detailed designs can be physical, conceptual, functional, or structural. Design knowledge includes, for example, traceability to requirements, standards, and specifications. Attributes of a part are described, as well as analyses, simulations, and trade studies. Other uses include validation/verification and operations/maintenance activities. Success in this evolution process depends on being able to capture "as-built" design knowledge from the outset. In a design knowledge environment "knowing why a quantity has a specific value is at least as important as knowing what the value is.[5]"

REFERENCES

1. Brachman, R.J., R.E. Fikes, and H.J. Levesque, KRYPTON: A Functional Approach to Knowledge Representation, IEEE Computer, Vol. 16(10), 1983, pp. 67-73.
2. NASA TM 89190, Advancing Automation and Robotics Technology for the Space Station, Advanced Technology Advisory Committee (ATAC), Progress Report Three, October 1, 1986, pp. 27-28.
3. Reiter, R., A Theory of Diagnosis from First Principles, Artificial Intelligence, Vol. 32, 1987, pp. 57-95.
4. Rowley, S., H. Shrobe, R. Cassels, and W. Hamscher, Joshua(TM): Uniform Access to Heterogeneous Knowledge Structures, AAAI-87, Proceedings of the Sixth National Conference on Artificial Intelligence, July 13-17, 1987, pp. 48-52.
5. Williams, B.C., Doing Time: Putting Qualitative Reasoning on Firmer Ground, AAAI-86, Proceedings of the Fifth National Conference on Artificial Intelligence, August 11-15, 1986, pp. 105-112.